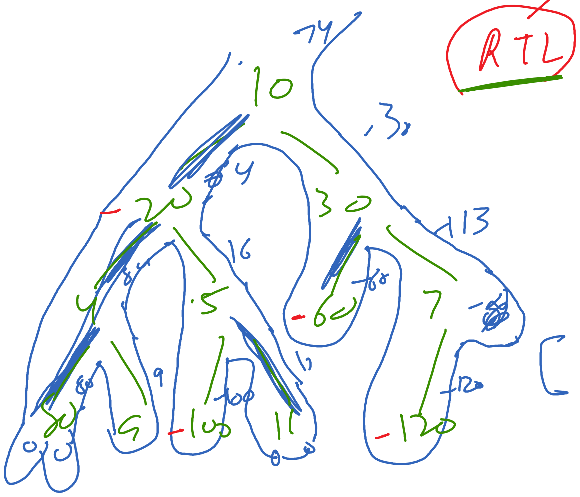
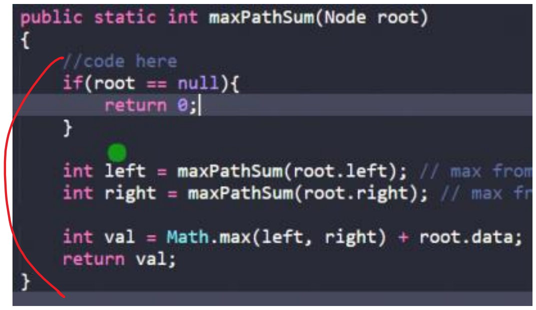
Tuesday, May 24, 2022 9:11 PM

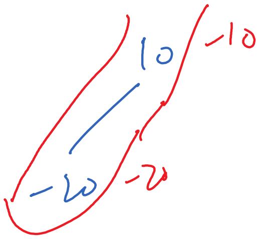
Max Path Sum [L T L]

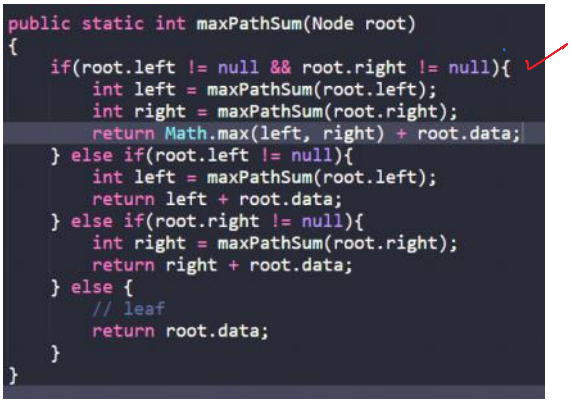
RTL RTN NTN



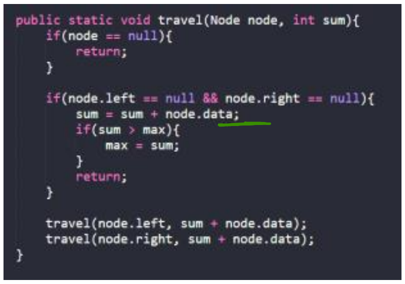


```
public static int maxPathSum(Node root)
{
    //code here
    if(root == null){
        return 0;
    }

    int left = maxPathSum(root.left); // max from
    int right = maxPathSum(root.right); // max fr

    int val = Math.max(left, right) + root.data;
    return val;
}
```

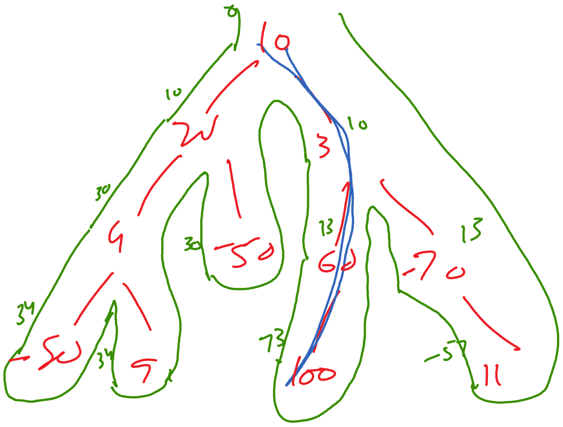




```
public static int maxPathSum(Node root)
{
    if(root.left != null && root.right != null){
        int left = maxPathSum(root.left);
        int right = maxPathSum(root.right);
        return Math.max(left, right) + root.data;
    } else if(root.left != null){
        int left = maxPathSum(root.left);
        return left + root.data;
    } else if(root.right != null){
        int right = maxPathSum(root.right);
        return right + root.data;
    } else {
        // leaf
        return root.data;
    }
}
```

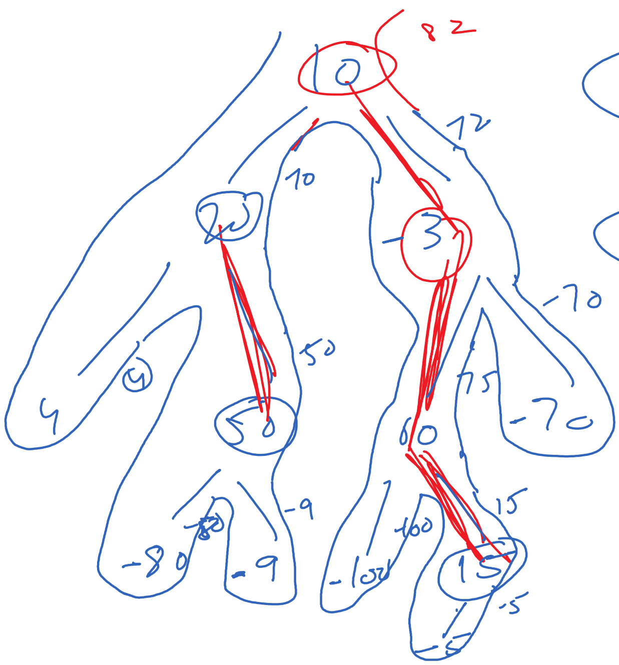


```
public static void travel(Node node, int sum){
    if(node == null){
        return;
    }

    if(node.left == null && node.right == null){
        sum = sum + node.data;
        if(sum > max){
            max = sum;
        }
        return;
    }

    travel(node.left, sum + node.data);
    travel(node.right, sum + node.data);
}
```



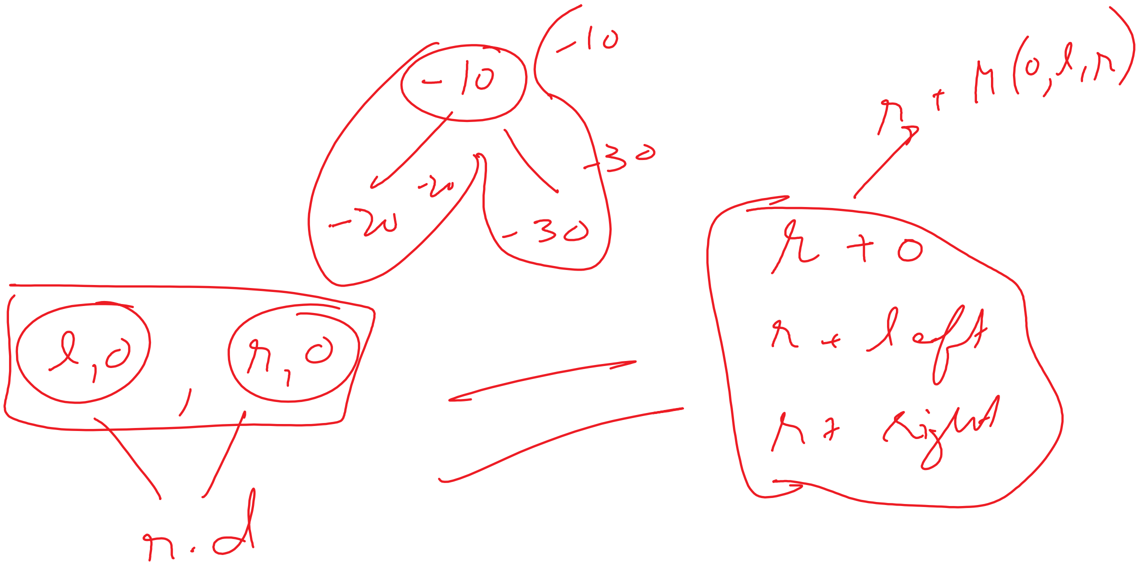
max = ~~-∞~~ ~~-54~~ ~~43~~ 173

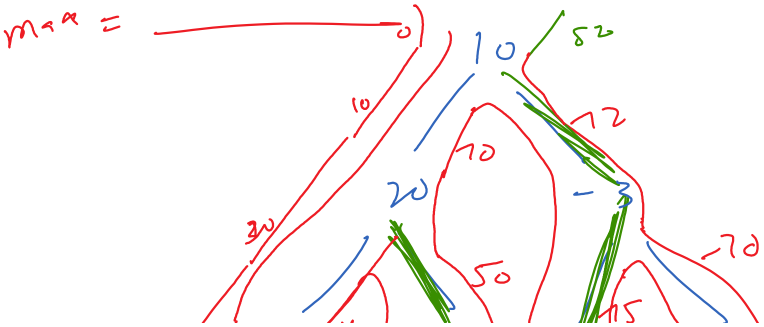
New Section 1 Page 1



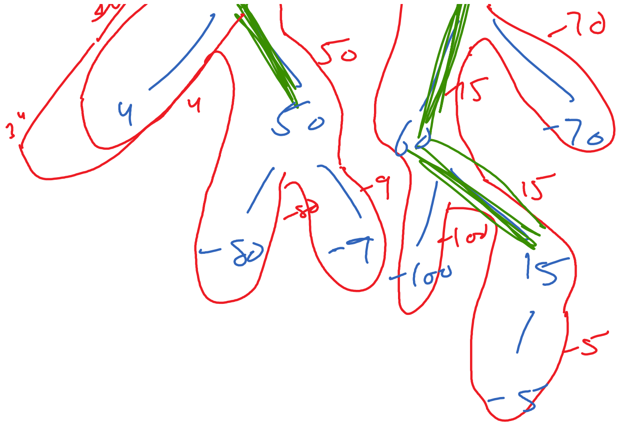
R to Node

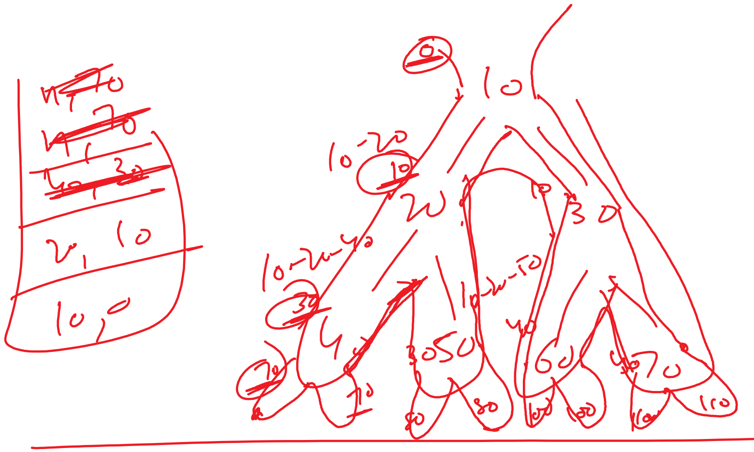
10:09 – 10:19





10

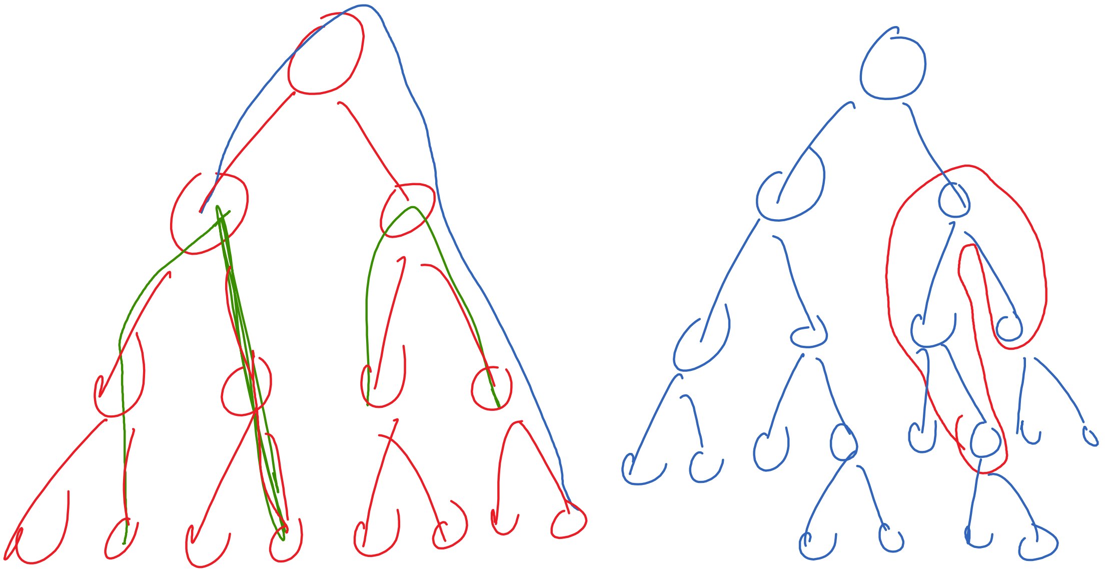




```
s2 = 0
trave (n, i sl) ?
    s2 += n.d;
    t (n.l, sl+n.d)
    t (n.r, sl+n.d)
```

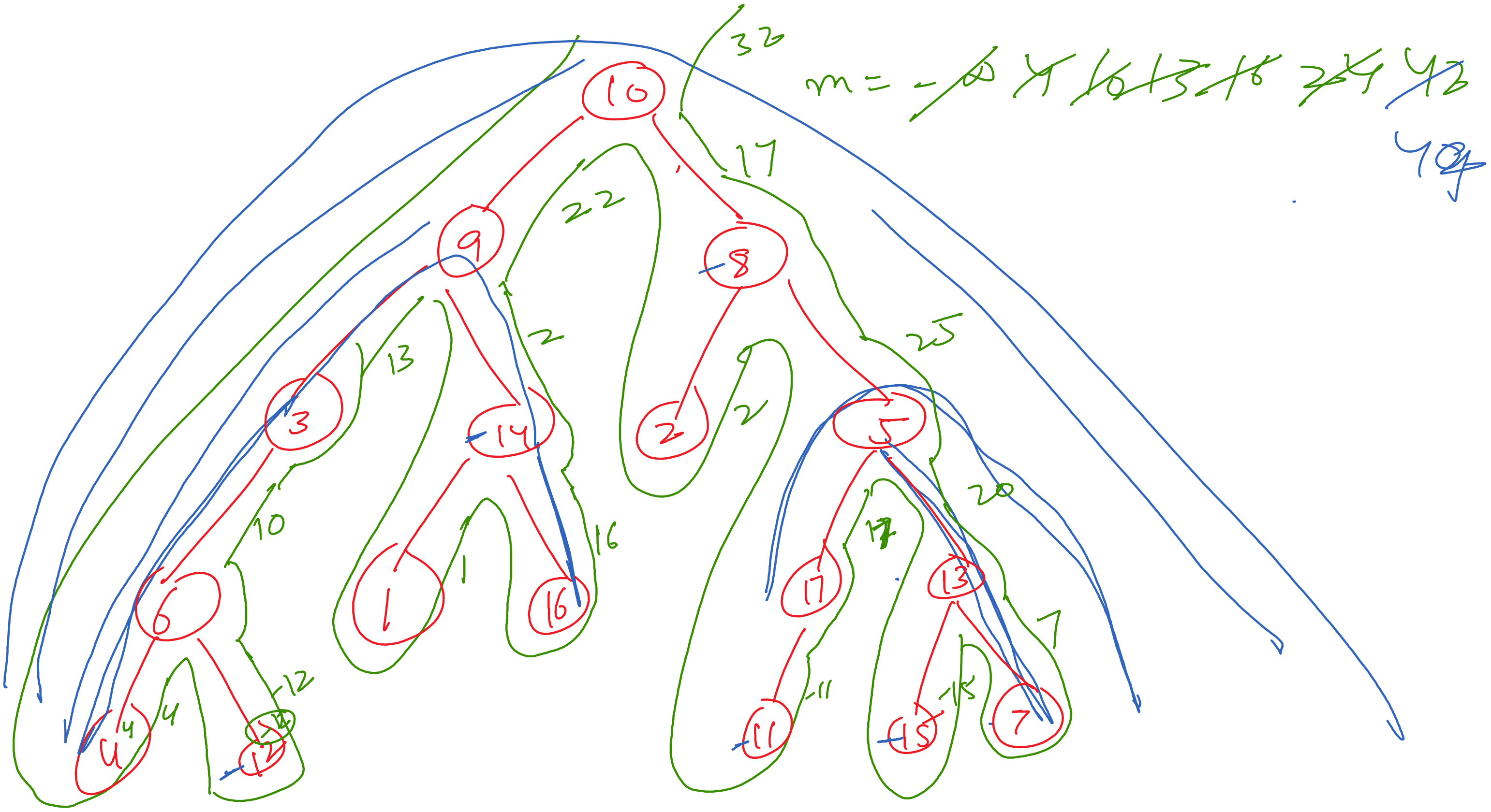
s

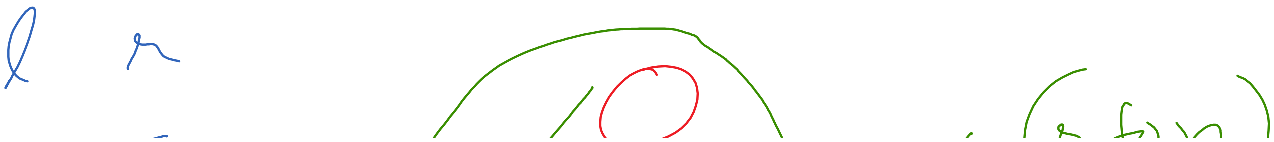
$s2 = \emptyset$ ~~10~~ ~~30~~ ~~70~~ ~~120~~ ~~150~~ ~~210~~ 280



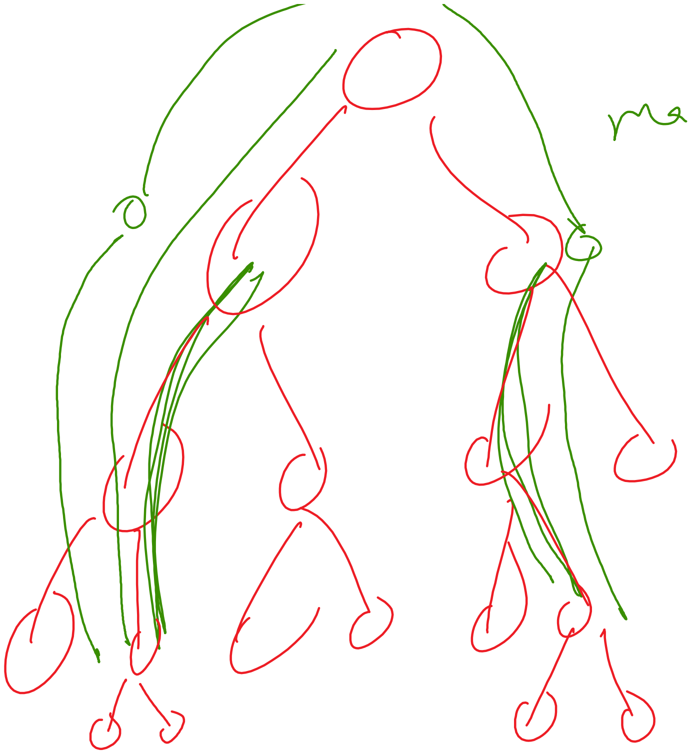
10:59 to 11:10

Max sum node to node



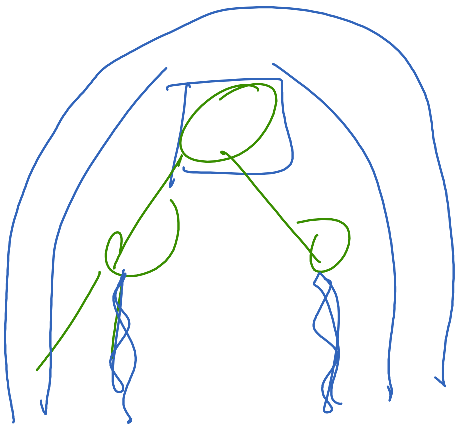


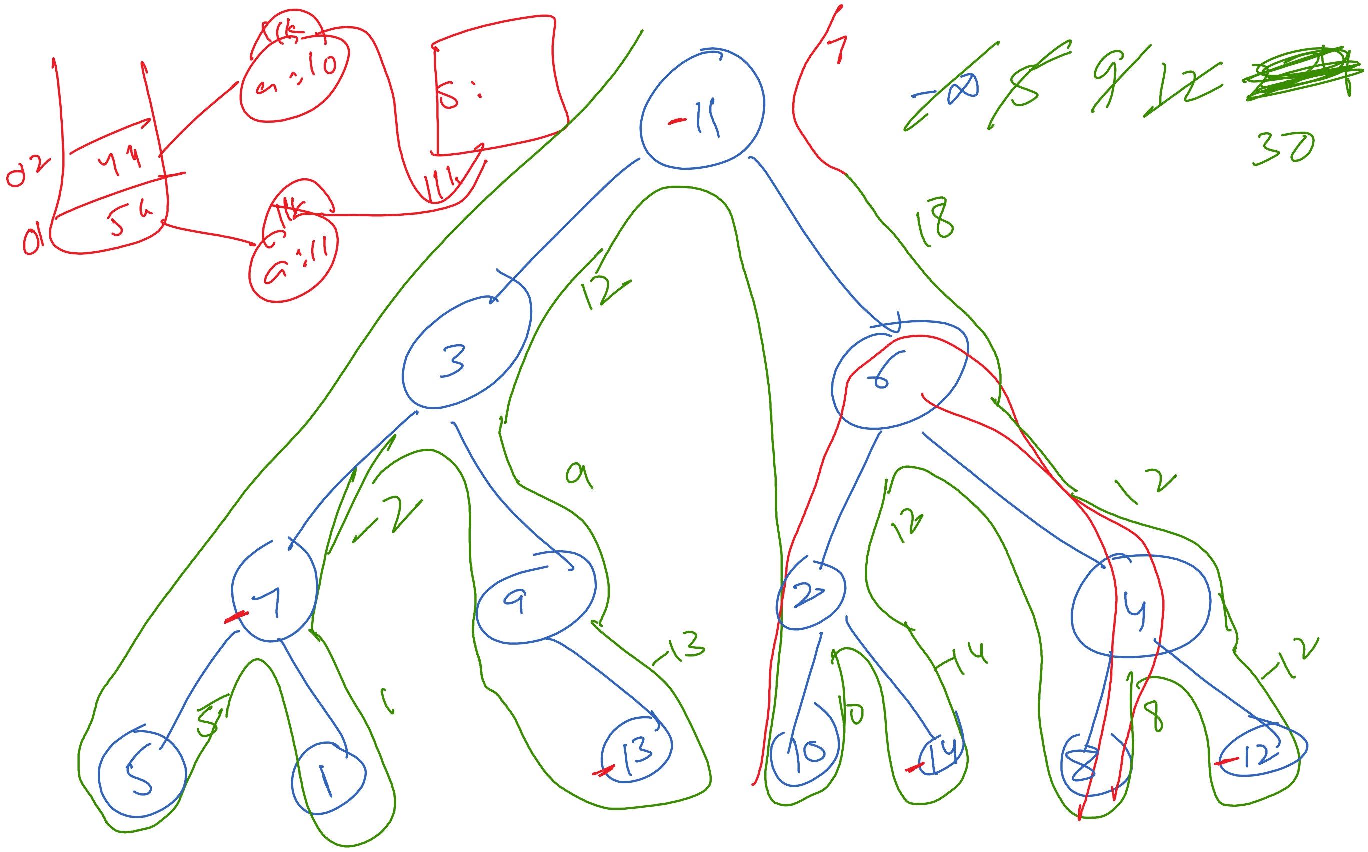
x	r
—	—
—	f
f	—
f	f



max (r to n)

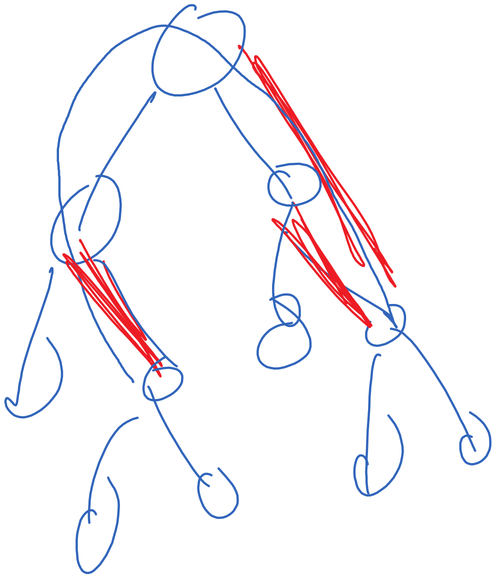
n to n

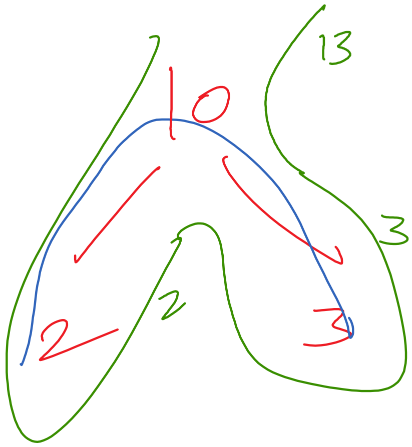


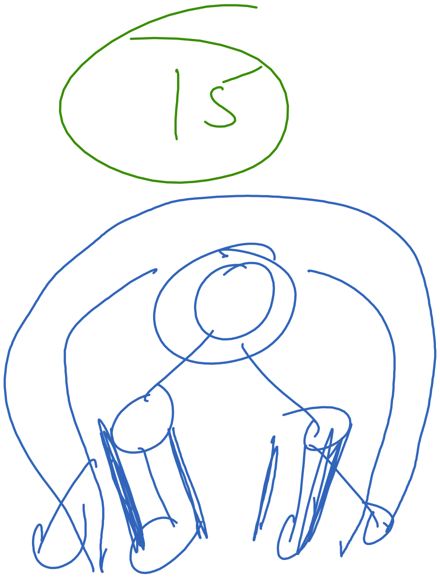


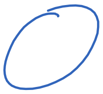
~~-∞~~ ~~8~~ ~~9~~ ~~12~~

30



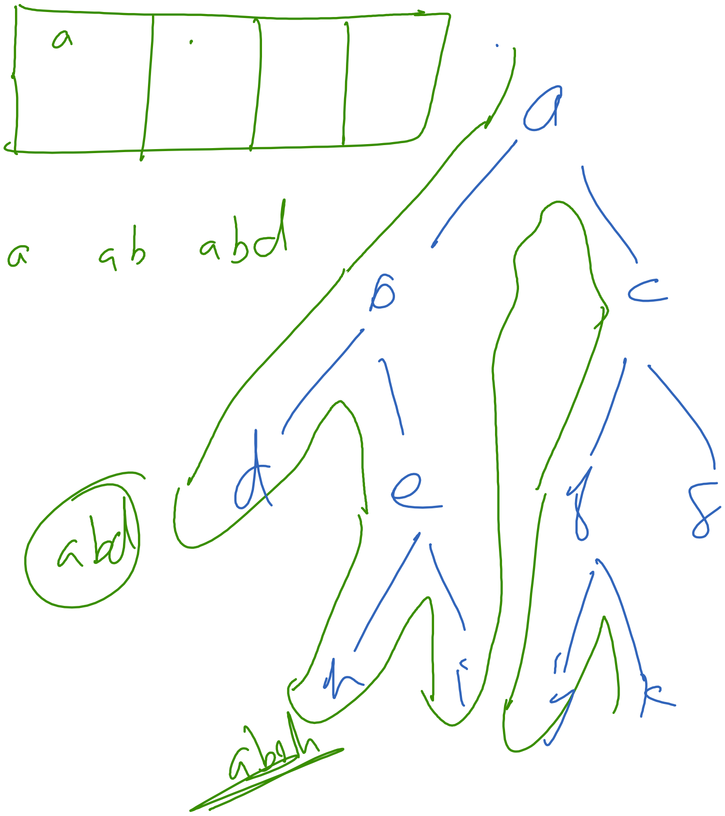






a			

a ab abd



abd

abbe

K

a b c d e f g 8

✓ ✓ ✓ ✓ ✓ ✓ ✓

2 5 -3 4 -6 8 7 2 -12 4

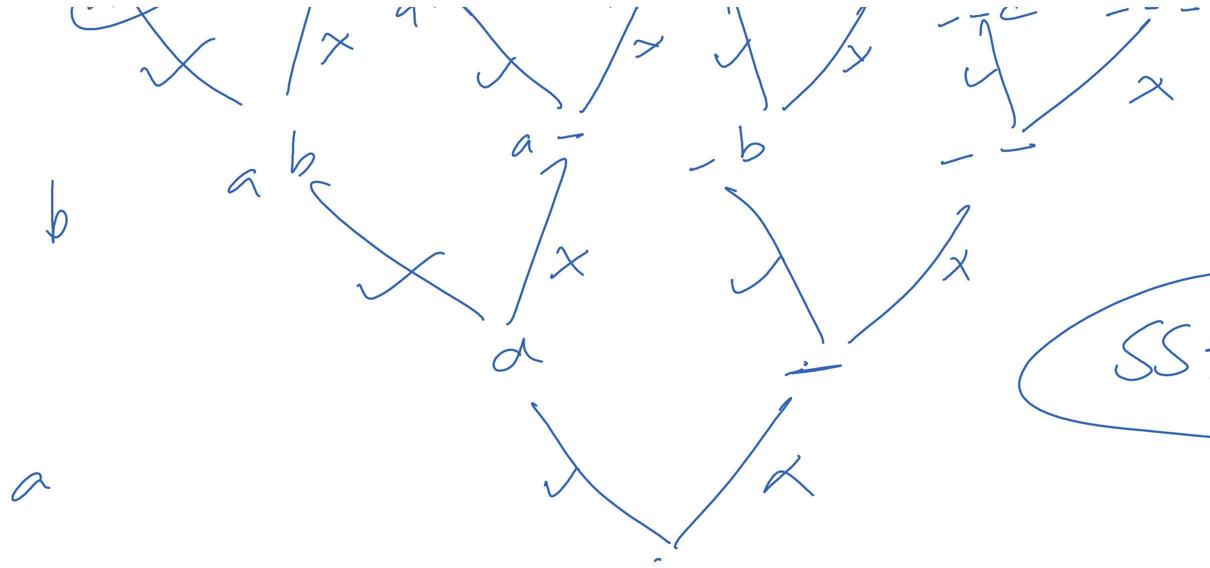
0	1	2	1
2	1	10	1
7	1		
4	1		
8	1		

2

Count of subarrays with sum equal to K

(abc ab- a-c a-- -bc -b- --c ---

✓ x ✓ x ✓ x ✓ x



SS = Perm